2014 Fourth International Conference on Advances in Computing and Communications

System Performance evaluation of Para virtualization, Container virtualization and Full virtualization using Xen, OpenVZ and XenServer

Anish Babu S*, Hareesh M J†, John Paul Martin‡, Sijo Cherian§ and Yedhu Sastri¶
Rajagiri School of Engineering and Technology
Rajagiri Valley, Kochi
India
*Email:anishsatheeshashish@yahoo.co.in
†Email:hareeshmjoseph@gmail.com
‡Email:johnpm12@gmail.com
§Email:sijoc@rajagiritech.ac.in
¶Email:yedhusastri@gmail.com

***Abstract*—When we consider virtualization, most of the times we get confused with virtualization technique that has to be used. That is which virtualization technology gives virtual machines with better system performance. The idea of this paper is to compare and evaluate the system performance of three basic hypervisors with one supporting Para virtualization other supporting Container virtualization and the last one supporting Full virtualization. Here we chose Xen-PV for Para Virtualization, OpenVZ for Container Virtualization and XenServer for Full virtualization. Then, performance of these three virtualization techniques is evaluated based on various tests conducted by the benchmarking tools. Finally a detailed comparison among these three hypervisors will be conducted to find out the hypervisor with which virtualization technique has better system performance.**

I. Introduction

In this paper mainly three kinds of virtualization technology are selected for evaluation. They are Para virtualization, Container virtualization and Full virtualization. These three are selected mainly because they are the common virtualization technologies in use now. The hypervisors selected here are the base hypervisors corresponding to each virtualization technology.

A. Para virtualization

Para virtualization[1] is a technique in which the guest operating system is aware that they are operating directly on the hypervisor instead of the underlying hardware. In Para virtualization a Para virtualization supporting hypervisor is installed on the host operating system which runs over the underlying hardware. This hypervisor will act as a virtualization layer. That is hypervisor will act as the host on which the guest operating systems are loaded. Guest operating systems will make the necessary system calls to hypervisors for the utilization of hardware resource. Typical examples include Xen-PV [4], ESXserver, KVM-PV etc. In this paper Xen-PV is used for testing Para virtualization and its performance.

1) Container virtualization: Container virtualization [2] is a technique in which each operating system kernel is modified to load multiple guest operating systems. Here guest operating systems are packed in the form of containers and each container will be allowed to load one by one. The kernel provides proper management of the underlying resources to isolate one container activity from the other. This type of virtualization technique has less overhead in loading the guest operating system in the form of containers and each container has their own IP address, memory, root access etc. In this paper OpenVZ [6] is used for testing and evaluating container virtualization.

B. Full virtualization

In full virtualization [3] hypervisor supporting the full virtualization technique is installed directly over the underlying hardware. This hypervisor is responsible for loading the guest operating systems. And each guest operating system will run as if they are operating directly on the underlying hardware. That is each guest operating system will get all the features of the underlying hardware. Here hypervisor will directly interact with the underlying memory and disk space and hypervisor will isolate the activities of one guest operating system from the other. Hypervisors supporting full virtualization have a virtual machine management console from which each guest virtual machines can be easily managed. Xenserver [5] is used here for testing and evaluating full virtualization.

This paper is organized in such a way that, in the next section detailed information about experimental phase is given. After that various results from the tests conducted will be plotted. Then finally a conclusion based on the results is provided.

II. Experiment Phase

We created a testing environment with similar hardware setup. That is machines with similar hardware configuration. For comparing Para virtualization Container virtualization and

978-1-4799-4363-0/14 $31.00 © 2014 IEEE
DOI 10.1109/ICACC.2014.66

Full virtualization, several hypervisors supporting Para virtualization, container virtualization and full virtualization is installed and configured on each machine. For Para virtualization the hypervisors we are using are Xen-PV. OpenVz is used to test container virtualization. XenServer are used to test Full virtualization. We installed these hypervisors in each machine and a virtual machine is loaded using these hypervisors. Debian wheezy is the operating system loaded using these hypervisors on each virtual machine. Then the performance of each hypervisors is evaluated based on the benchmark values generated by the UnixBench bench marking tool.Unix bench is used for evaluating the system performance of the virtual machines loaded with Unix based operating system. Multiple tests are conducted on the virtual machines to test various features of the system performance. These test results are then compared with a base score to generate the index value. High index value in a test means better performance for that particular test.Various tests include string handling capability, speed and efficiency of floating point operations, number of execl calls per second, file copy rates, rate at which a process can write bytes into pipe, rate at which two processes can exchange information through pipe, Process creation, shell script rate, system call overhead etc.

We loaded virtual machines with Debian operating system using each hypervisors. Then run unixBench evaluation on each virtual machines to analyze its system performance. Then a detailed comparison based on each test result is performed. At last a final index value for Para virtualization, Container virtualization and Full virtualization is calculated based on the test results and various graphs are also plotted based on the test results.

III. Results

The results we obtained are plotted as the graphs as shown below.

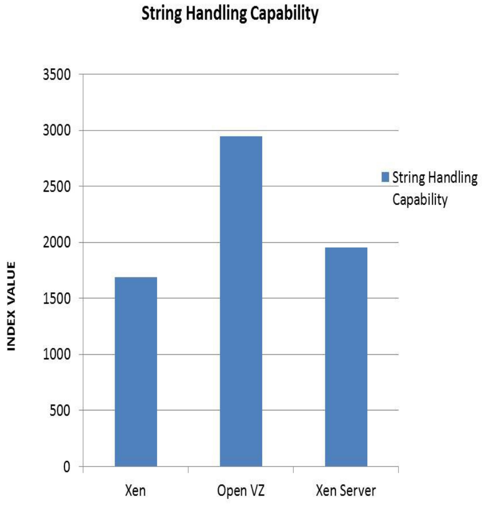

Fig. 1: Time versus Replicating-Packets Graph

The above graph shows the index value generated by UnixBench for string handling. From this graph OpenVz has high index value than that of the other two. From the graph it is clear that the hypervisor supporting Container virtualization have higher index value for string handling than the other two.

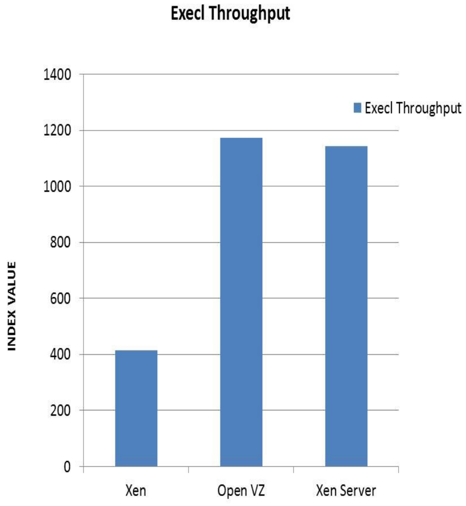

Fig. 2: Time versus Replicating-Packets Graph

From the graph in figure 2 it is clear that Execl throughput index value is slightly greater for container virtualization than the other two. That is number of excel call per second is higher for OpenVZ than the other two. From the graph it is clear that the hypervisor supporting Container virtualization has slightly higher Execl throughput index value than the other two.

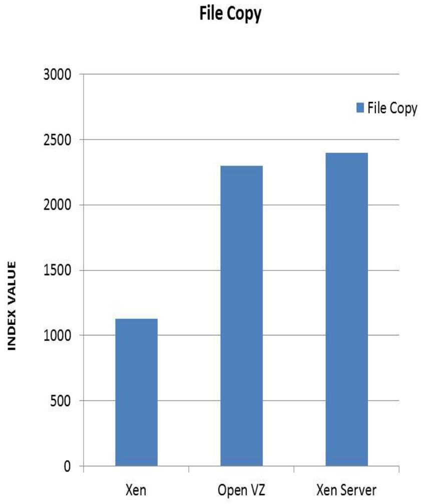

Fig. 3: Time versus Replicating-Packets Graph

From the graph in figure 3, it is clear that XenServer has a higher index value for File copy than the other two. That is Full virtualization has high file copy rate than the other two.The rate at which data can be transferred from one file to other using different buffer size is greater for XenServer. From the graph it is clear that hypervisor supporting full virtualization haS slightly higher index value for file handling than the other two.

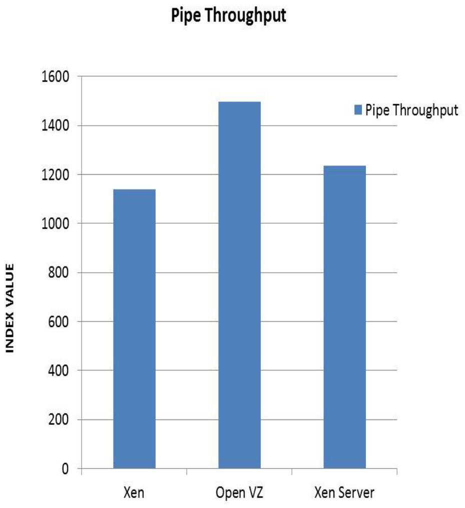

Fig. 4: Time versus Replicating-Packets Graph

Fig 4 shows the index value generated by Unix Bench for OpenVz is higher for Pipe throughput. Pipe throughput indicates the number of times a process can write certain amount of data into the pipe and read them back. From the graph, hypervisor supporting Container virtualization has slightly higher index value for pipe throughput than the other two.

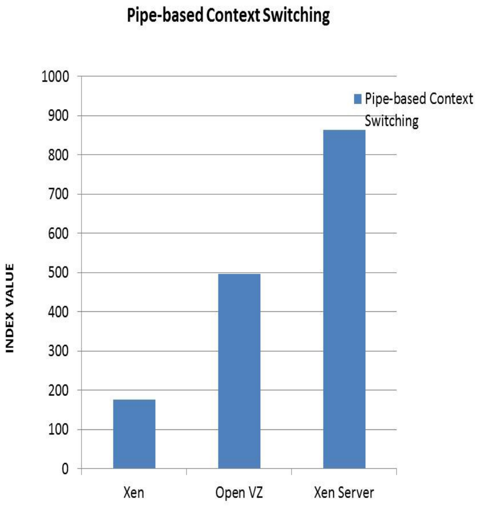

Fig. 5: Time versus Replicating-Packets Graph

From fig 5 it is clear that Index value for Pipe-based Context switching is very much higher for XenServer than the other two. This test the measures the number of times an increasing integer is exchanged between two processes through the pipe. The graph gives the idea that the hypervisor supporting full virtualization has a very high index value for pipe based context switching than the other two.

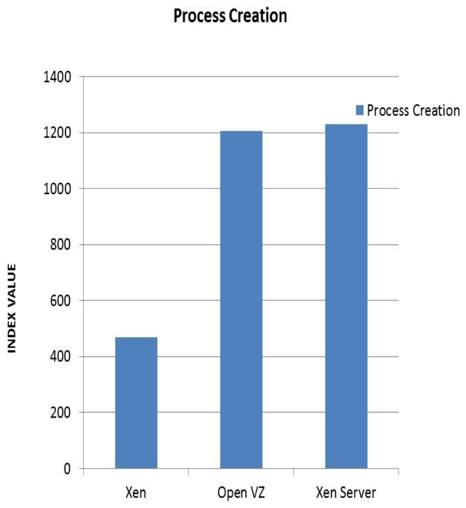

Fig. 6: Time versus Replicating-Packets Graph

From fig 6 Xenserver is slightly more advantageous than the other two. This test is mainly used to compare various implementations of operating system process creation calls. That is process creation and allocation of memory to each process is included here. From the graph it is clear that hypervisor supporting full virtualization have slightly higher index value for process creation than the other two.

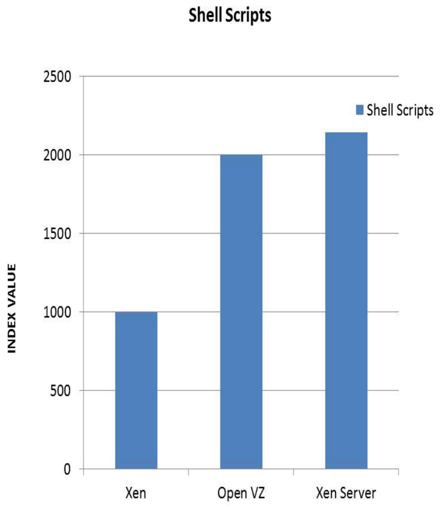

Fig. 7: Time versus Replicating-Packets Graph

From fig 7, we get the idea that Xenserver has higher index value for shellscripts than the other two. This test measures the number of times a process can start and execute shell scripts that make changes in data. From the graph it is clear that the hypervisor supporting full virtualization has higher index value for shell scripts than the other two.

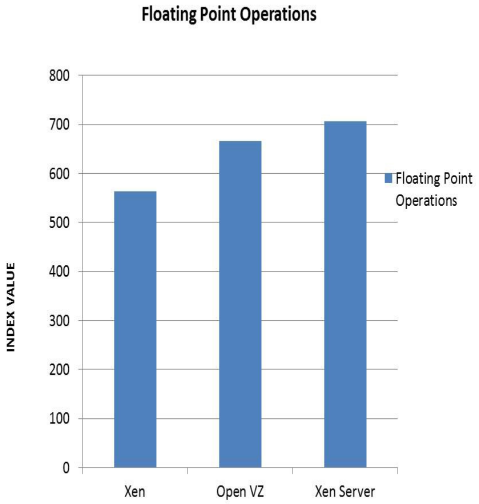

Fig. 8: Time versus Replicating-Packets Graph

Fig 8 explicitly give the idea that Xenserver has high index value for floating point operations than the other two. This test measures the speed and efficiency of floating point operations. From the graph it is clear that the hypervisor supporting full virtualization has higher index value for Floating point operations than the other two.

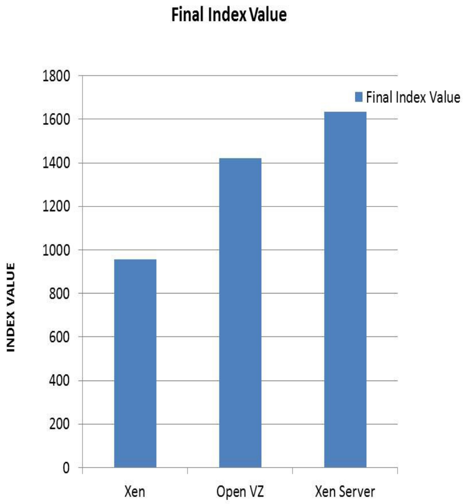

Fig. 9: Time versus Replicating-Packets Graph

From the graph in fig 9 with final index scores, XenServer has higher system performance compared to OpenVZ and Xen-PV. System performance index value is calculated based on the values of the aforementioned tests. That is Virtual machines loaded with the hypervisor supporting full virtualization have better system performance than Container virtualization and Para virtualization.

IV. Conclusion

A detailed comparison of the three hypervisors supporting three different virtualization techniques is done. This comparison is performed based on several tests that are conducted based on the system performance of the virtual machines loaded by the hypervisors. From the results obtained we can see that different hypervisors show different performance in different tests conducted. For some tests OpenVZ gives high performance but for majority of the tests XenServer gives high performance. Xen supporting Para virtualization shows less performance in all tests. Thus from the hypervisors analyzed in this paper, the hypervisor with full virtualization support has comparatively higher system performance in terms of File copy, Pipe based context switching, Process creation, Shell scripts and Floating point operations than the hypervisors supporting Para virtualization and Container virtualization.

References

[1] Lei Yu, Chuliang Weng, Minglu Li, and Yuan Luo, *SNPdisk: An Efficient Para-Virtualization Snapshot Mechanism for Virtual Disks in Private Clouds*, IEEE Network July/August 2011.

[2] Miguel G. Xavier, Marcelo V. Neves, Fabio D. Rossi, Tiago C. Ferreto, Timoteo Lange, Cesar A. F. De Rose, *Performance Evaluation of Container-based Virtualization for High Performance Computing Environments*, 21st Euromicro International Conference on Parallel, Distributed, and Network-Based Processing IEEE 2013.

[3] Wei Chen, Hongyi Lu, Li Shen, Zhiying Wang, Nong Xiao, Dan Chen, *A Novel Hardware Assisted Full Virtualization Technique*, The 9th International Conference for Young Computer Scientists IEEE 2008.

[4] Igli TAFA, Ermal BEQIRI, Hakik Paci, Elinda KAJO, Aleksandr XHUVANI, *The evaluation of Transfer Time, CPU Co nsumption and Memory Utilization in XEN-PV, XEN-HVM, OpenVZ, KVMFV and KVM-PV Hypervisors using FTP and HTTP approaches* , Third International Conference on Intelligent Networking and Collaborative Systems IEEE 2011.

[5] Michael Terrell, Natarajan Meghanathan, *Setting up of a Cloud Cyber Infrastrcture using Xen Hypervisor*, 10th International Conference on Information Technology: New Generations IEEE 2013

[6] A. Kovari, P. Dukan, *KVM OpenVZ virtualization based IaaS Open Source Cloud Virtualization Platforms: OpenNode, Proxmox VE*, 10th Jubilee International Symposium on Intelligent Systems and Informatics, IEEE 2012